



Fernando, WAC., Canagarajah, CN., & Bull, DR. (2000). Fade-in and fade-out detection in video sequences using histograms. In *IEEE Int. Symposium on Circuits and Systems (ISCAS), Geneva* (Vol. 4, pp. 709 - 712). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/ISCAS.2000.858850>

Peer reviewed version

Link to published version (if available):  
[10.1109/ISCAS.2000.858850](https://doi.org/10.1109/ISCAS.2000.858850)

[Link to publication record in Explore Bristol Research](#)  
PDF-document

## University of Bristol - Explore Bristol Research

### General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:  
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

## FADE-IN AND FADE-OUT DETECTION IN VIDEO SEQUENCES USING HISTOGRAMS

W.A.C. Fernando, C.N. Canagarajah, D. R. Bull

Image Communications Group, Centre for Communications Research,  
University of Bristol, Merchant Ventures Building, Woodland Road,  
Bristol BS8 1UB, United Kingdom.

Voice - +44-117-954-5198, Fax - +44-117-954-5206

Email - W.A.C.Fernando@bristol.ac.uk

### ABSTRACT

There is an increased need to extract key information automatically from video for the purposes of indexing, fast retrieval, and scene analysis. To support this vision, reliable scene change detection algorithms must be developed. Several algorithms have been proposed for both sudden and gradual scene change detection in uncompressed and compressed video [1-8]. In this paper we present an algorithm for fade-in and fade-out scene change detection in both uncompressed and compressed video sequences using histograms. We use the properties of the fading operation and extract these features in the luminance histogram. Results show that the proposed algorithm can be used in both uncompressed and compressed video to detect fade regions with a high reliability and less computational computations.

### 1. INTRODUCTION

A major feature required in a visual information system is an efficient indexing system to enable fast access to the stored data. A video index serves as a descriptor of the video, thus enabling fast access to the video clips stored in a multimedia databases. Another consideration is that a user who is interested in searching, browsing or retrieving video clips needs a way to interface with the database by formulating appropriate queries. These queries need to be appropriately translated into a form that can be used to search an index and retrieve the matching clips. A typical approach to indexing and archiving video for retrieval requires parsing the video, extracting key information from each clip, indexing the information and providing a representation which allows accurate and efficient retrieval based on the user's request. The large channel bandwidth and memory requirements for the transmission and storage of image and video necessitate the use of video compression techniques. Hence, the visual data in multimedia databases is expected to be stored mostly in the compressed form. In order to avoid the unnecessary decompression operations in indexing and searching processes, it is efficient to index the image and video in the compressed format itself. Therefore, video indexing demand a powerful scene change detection algorithm to allow for a complete characterisation of the video sequences.

Scene changes (transitions) can be divided into two categories: abrupt transitions and gradual transitions. Gradual transitions

include camera movements: panning, tilting, zooming and video editing special effects: fade-in, fade-out, dissolving, wiping. Fade-in and fade-out operations are very common in TV programme production. In this paper we present a simple algorithm for fade-in and fade-out scene change detection in both uncompressed and compressed video sequences using information in the luminance histogram. We use the properties of the fading operation and extract these features in the luminance histogram. In compressed domain we use DC-sequences [3] to construct the reduced index image and apply the same criteria as in uncompressed domain on this reduced image.

Rest of the paper is organised as follows: We review briefly in section 2 several related works on scene change detection in uncompressed and compressed video. Section 3 presents a mathematical model for fading operations in video production and how it effects for the histogram. This section also describes the implementation of the proposed scheme in uncompressed video and MPEG-2 compressed video. Some experimental results are presented in section 4. Section 5 discusses the conclusions and future work.

### 2. RELATED WORK

Abrupt transitions are very easy to detect as the two frames are completely uncorrelated [1,2]. Gradual transitions are more difficult to detect as the difference between frames corresponding to two successive shots is substantially reduced. However, several algorithms have also been proposed for gradual scene change detection [3-8]. Zabith proposed a feature-based algorithm for detecting fading [4]. This algorithm needs to detect edges in every frame and hence it is very costly. Another limitation of this scheme is that the edge detection method does not handle rapid changes in overall scene brightness, or scenes with a high contrast. Furthermore, automatic segmentation and classification is not possible with this scheme. Alattar proposed an algorithm for fade region detection by exploiting the semi-parabolic behaviour of the variance curve [5]. This algorithm can only detect fade-in and fade-out when the end frames are fixed (frozen). When the sequence has considerable motion, this algorithm fails to identify fade-in and fade-out regions. We have proposed an algorithm for detecting fade-in and fade-out using statistical features of both luminance and chrominance signals [6]. We considered the ratio between incremental change in the mean

of the luminance signal to the same for the chrominance (average sum of  $C_r$  and  $C_b$ ) signal to identify fade-in and fade-out. However, this algorithm may fail to identify fading regions when the solid colour is very close to the mean of the original sequence (before fading is applied). Main common drawback of these algorithms is that all these schemes can work only in uncompressed domain. Since video is typically stored in a compressed format, fading detection in compressed domain is also essential. Therefore, an algorithm, which can operate within a single framework for both compressed and uncompressed domain, would be highly desirable for content-based video.

## 2. PROPOSED SCHEME

### 3.1 Mathematical Model for Fading

In video editing and production, proportions of two or more picture signals are simply added together so that the two pictures appear to merge on the output screen. Very often this process is used to move on from picture A to picture B. In this case, the proportions of the two signals are such that as the contribution of picture A changes from 100% to zero, the contribution of picture B changes from zero to 100%. This is called dissolving. When picture A is a solid colour, it is called as fade-in and when picture B is a solid colour, it is known as fade-out. Mathematically, fade-in and fade-out can be modelled as shown in Equations (1) and (2) respectively.

$$S_n(i, j) = \begin{cases} f_n(i, j) & 0 \leq n < L_1 \\ \left[1 - \left(\frac{n - L_1}{F}\right)\right]C + \left(\frac{n - L_1}{F}\right)g_n(i, j) & L_1 \leq n \leq (L_1 + F) \\ g_n(i, j) & (L_1 + F) < n \leq L_2 \end{cases} \quad (1)$$

$$S_n(i, j) = \begin{cases} f_n(i, j) & 0 \leq n < L_1 \\ \left[1 - \left(\frac{n - L_1}{F}\right)\right]f_n(i, j) + \left(\frac{n - L_1}{F}\right)C & L_1 \leq n \leq (L_1 + F) \\ g_n(i, j) & (L_1 + F) < n \leq L_2 \end{cases} \quad (2)$$

where,  $C$  is the video signal level (solid value),  $S_n(i, j)$  is the resultant video signal,  $f_n(i, j)$  is picture A,  $g_n(i, j)$  is picture B,  $L_1$  is length of sequence A,  $F$  is length of fading sequence,  $L_2$  is length of the total sequence.

### 3.2 Histograms for Fading Identification

Histogram of the luminance signal is a well established technique to identify sudden scene changes in uncompressed domain [2,7]. The histogram is obtained from the number of pixels belonging to each gray level in the frame. Let the image

size be  $(M, N)$  and each luminance pixel is coded with eight bits. Then the histogram has 256 bins and  $M * N$  number of votes altogether. Bins are voted according to the value of each pixel in the image.

In this section we focus on properties of the histograms during the fading operation. Let,  $H_{S,n}(x)$ ,  $H_{f,n}(x)$ , and  $H_{g,n}(x)$  define the histograms for  $S(x, y)$ ,  $f(x, y)$  and  $g(x, y)$  respectively. Then the histogram of the resultant video signal can be defined as in Equation (3).

$$H_{S,n}(x) = \begin{cases} H_{f,n}(x) & 0 \leq n < L_1 \\ \left[1 - \left(\frac{n - L_1}{F}\right)\right]H_{f,n}\left(x - \left(\frac{n - L_1}{F}\right)C\right) & L_1 \leq n \leq (L_1 + F) \\ H_{g,n}(x) & (L_1 + F) < n \leq L_2 \end{cases} \quad (3)$$

Thus it is clear that during a fade-out operation, histogram is scaled and shifted along x-axis according to the fading rate. We used this scaling property of the histograms to identify fading transitions. We calculate the horizontal span  $X(n)$  of the histogram and monitor the changes. During fading operation  $X(n)$  should be decrease or increase with the frame number for fade-out and fade-in operations respectively. Figure 1-3 show how the horizontal span is changing for a typical fade-out operation. To improve the reliability of the algorithm, we can consider the colour histograms as well. But, this will increase the computational complexity in compressed domain. Therefore, we divide the luminance image into number of regions and consider the histograms for these regions separately. If the number of regions are small, reliability of the algorithm may reduce. If the number of regions are high, computational cost increases. Therefore, number of regions are fixed to four to optimise these two scenarios. Thus, in the proposed algorithm, we divide each image into four segments as shown in Figure (4). If we construct four histograms for these four segments, they should also be scaled when there is a fading operation. We use these four histograms to identify fading in the video sequence.

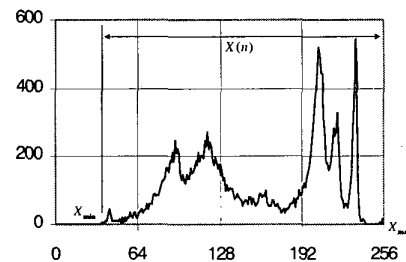


Figure 1: Histogram at the start of fade-out operation

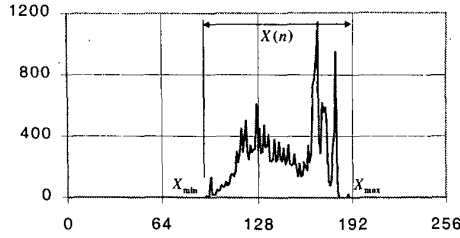


Figure 2: Histogram at the middle of fade-out operation

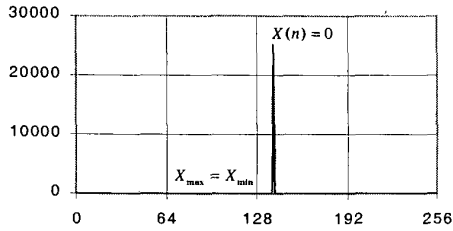


Figure 3: Histogram at the end of fade-out operation

1	$X_{1,\min}$ $X_{1,\max}$	2	$X_{2,\min}$ $X_{2,\max}$
3	$X_{3,\min}$ $X_{3,\max}$	4	$X_{4,\min}$ $X_{4,\max}$

Figure 4: Four segments of the image and their parameters

Horizontal spans can be calculated by searching the maximum and the minimum value of the luminance pixel in every image of the sequence as shown in Figure (4). Thereby, we can eliminate the computational cost for constructing the histograms. Then, we compare the adjacent three values of horizontal spans to identify the start frame of fade-out operation. This condition is explained in Equation (5). Finally, we consider all four segments to make sure that fade-out is started actually at frame "n" as shown in Equation (6). If  $C(n) = 1$ , then we declare that fade-out has started at frame "n". The reason to consider three adjacent frames is to compromise the reliability and the delay. If we consider more frames reliability of the algorithm improves but the delay increases.

$$X_i(n) = X_{i,\max}(n) - X_{i,\min}(n) \quad (4)$$

$$Y_i(n) = [X_i(n-3) > X_i(n-2)] \text{AND} [X_i(n-2) > X_i(n-1)] \text{AND} [X_i(n-1) > X_i(n)] \quad (5)$$

where,  $i = 1, 2, 3, 4$

$$C(n) = [Y_1(n)] \text{AND} [Y_2(n)] \text{AND} [Y_3(n)] \text{AND} [Y_4(n)] \quad (6)$$

Finally, last frame of fade-out operation can be identified by checking the value of  $X(n)$ . At the end of fade-out operation,  $X(n)$  should be zero. Similar arguments can be followed to identify fade-in regions. Figure 5 illustrates the complete algorithm for detection of fade-out and fade-in regions. Here,  $D(n)$  is defined according to Equation (7) and (8).

$$Z_i(n) = [X_i(n-3) < X_i(n-2)] \text{AND} [X_i(n-2) < X_i(n-1)] \text{AND} [X_i(n-1) < X_i(n)] \quad (7)$$

$$D(n) = [Z_1(n)] \text{AND} [Z_2(n)] \text{AND} [Z_3(n)] \text{AND} [Z_4(n)] \quad (8)$$

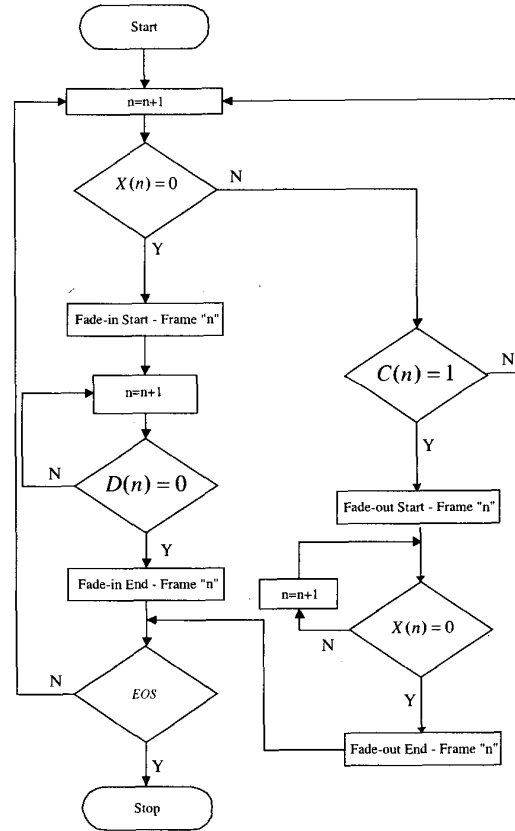


Figure 5: Flow chart of the proposed algorithm (EOS – End of sequence)

There may be a possibility of satisfying this condition for a small number of consecutive frames in a non-fade sequence. But, small sizes of fade sequences are not common in practice. Using this argument false regions are eliminated in the proposed algorithm. Furthermore, if there are two consecutive fade regions separated by very small gap, they are bridged to form a longer fade region.

### 3.3 Detection of Fading in MPEG-2 Compressed Video

This algorithm can also be applied in compressed domain by considering DC-sequences [3] of each frame. In compressed domain, DC values of each 8x8 macroblock are calculated using DC-sequences. Therefore, we can use this mean value to represent the 8x8 pixel values of the original image. Then we have a reduced image ( $M/8, N/8$ ) to apply the same algorithm explained in section 3.2 to identify fade regions in MPEG-2 compressed video.

## 4. RESULTS

We considered a video sequence of 2200 frames to test the proposed algorithm. This sequence contains several other special effects such as wiping, panning and dissolving. Same sequence is applied to MPEG-2 encoder and compressed sequence is analysed with the proposed algorithm in section 3.3. Table 1 shows the summarised results with the proposed algorithm in both uncompressed and compressed domain. Results show that the algorithm is capable of detecting all fade regions accurately even when the video sequence contains other special effects.

Actual fade region	Detected fade region (Uncompressed)	Detected fade region (Compressed)	Nature of the region
31-60	31-60	31-60	fade-in
111-150	111-150	111-150	fade-out
248-303	248-303	248-303	fade-out
576-624	576-624	576-624	fade-in
754-778	754-778	754-778	fade-out
944-986	944-986	944-986	fade-in
1102-1167	1102-1167	1102-1167	fade-in
1365-1420	1365-1420	1365-1420	fade-out
1500-1550	1500-1550	1500-1551	fade-in
1620-1680	1620-1680	1620-1680	fade-in
1760-1840	1760-1840	1760-1840	fade-out
1920-1985	1920-1985	1920-1985	fade-out

**Table 1:** Fade region identification with the proposed algorithm

Main advantage of the proposed scheme is that it is simple and computationally inexpensive. Histograms have been used as a reliable technique for sudden scene change detection in uncompressed domain [2,7]. Therefore, this proposed fade detection algorithm can easily be combined with the sudden scene change detection algorithm. Thus, it can be used to identify both sudden scene changes and fading. Therefore, the proposed algorithm can be used in both uncompressed and compressed video to detect fade regions with a high reliability with less computational computations.

## 5. CONCLUSIONS

In this paper, we presented an algorithm for fade-in and fade-out scene change detection in both uncompressed and

compressed video sequences using histograms. We used the properties of the fading operation and extract these features in the luminance histogram. Same algorithm is used with DC-sequences [3] to identify these transitions in MPEG-2 compressed video. Results show that the proposed algorithm can be used in both uncompressed and compressed video to detect fade-in and fade-out with a very high reliability rate. Unlike in other algorithms [5-6], this proposed scheme is simple and computationally inexpensive. However, future work is required to extend this work to identify dissolving in uncompressed and compressed video.

## ACKNOWLEDGEMENTS

First author would like to express his gratitude and sincere appreciation to the University of Bristol and CVCP for providing financial support for this work.

## REFERENCES

1. Fernando, W.A.C., Canagarajah, C.N., Bull, D. R., "Sudden Scene Change Detection in MPEG-2 Video Sequences," Proceedings - IEEE International Workshop on Multimedia Signal Processing, pp. 259-264, 1999.
2. Zhang, H.J., "Automatic Partitioning of Full-Motion Video," ACM/Springer Multimedia Systems, Vol.1, No.1, pp. 10-28, 1993.
3. Yeo, B.L., "Rapid Scene Analysis on Compressed Video," IEEE Transactions on circuits and systems for video technology, Vol. 5, No 6, pp. 533-544, December 1995.
4. Zabith, R., Miller, J., and Mai, K., "Feature-Based Algorithms for Detecting and Classifying Scene Breaks," 4<sup>th</sup> ACM Intl. Conference on Multimedia, San Francisco, California, November 1995.
5. Alattar, A.M., "Detecting Fade Regions in Uncompressed Video Sequences," pp. 3025-3028, ICASSP, 1997.
6. Fernando, W.A.C., Canagarajah, C.N., Bull, D. R., "Automatic Detection of Fade-in and Fade-out in Video Sequences," Volume IV, ISCAS, pp. 255-258, 1999.
7. Nagasaka, A., Tanaka, Y., "Automatic Video Indexing and Full-Video Search for Object Appearances," Visual Database Systems II, Eds.E.Kunth, and L.M. Wegner, Elsevier Science Publishers B.V., IFIP, pp. 113-127, 1992.
8. Fernando, W.A.C., Canagarajah, C.N., Bull, D. R., "Video Segmentation and Classification for Content Based Storage and Retrieval Using Motion Vectors," Storage and Retrieval for Image and Video Databases VII, SPIE, pp. 687-698, 1999.